

Appendix - Whitman College Islandora 2.0 Migration Guide

- [General Workflow](#)
- [Provision a Local Environment](#)
- [Determine File Location](#)
- [Create a Collection in Islandora 2.0](#)
- [Get the CSV File](#)
- [Prepare Config File](#)
- [Prepare CSV File](#)
 - [CSV Required Fields](#)
 - [Title](#)
 - [Resource Type](#)
 - [System Model](#)
 - [File](#)
 - [Other General CSV Notes](#)
 - [Delimiter](#)
 - [Subdelimiter](#)
 - [Term Creation](#)
 - [Fields Created by Default](#)
 - [Adding New Fields](#)
 - [Special Fields and Field Types](#)
 - [Linked Agent](#)
 - [Adding a Custom Relation](#)
 - [EDTF Formats](#)
 - [Links](#)
 - [Coordinates](#)
 - [Member Of](#)
 - [Display Hints](#)
 - [Configure the CSV](#)
 - [Configuring Complex Objects in the CSV](#)
 - [Newspapers](#)
 - [Books](#)
 - [Compound Objects](#)
- [Check, Then Run](#)
- [Quality Assurance/Quality Control post-ingest](#)
- [Appendix A: Default Relationship Types](#)
- [Appendix B: Born-Digital i8 Theme Object View Configurations](#)

This document outlines the migration process followed by Whitman College. It is based on Islandora Workbench and the theme contributed to the Islandora Foundation by Born Digital. Some elements may need to be altered in order for this process to work with a different theme.

General Workflow

1. Build a collection on Islandora 2.0 server to accept inputs. (From the browser interface)
2. Build a config file with defaults to cover accessibility, authorship, access, etc.
3. Verify the input spreadsheet - make sure column headers have valid fieldnames, and build URLs from pids if necessary (an easy macro in the spreadsheet).
4. Dry run, then run. (Both from the command line.)
5. Check results, then accept, or rollback as necessary.
6. Add the spreadsheet to the input archive.

Provision a Local Environment

If you have not already done so, provision a local Islandora 2.0 environment to use for the ingests. We recommend using ISLE with the codebase/sandbox option. Running the command: `starter_dev` will bring up the site in the preferred theme. More complete ISLE documentation for starter instances can be found on the [ISLE wiki here](#).

Determine File Location

Workbench can be configured to retrieve source files either by URL or from a locally available directory. In the former case, file URLs must be included in a column in the CSV, while for the later case the location of the directory is specified in the config file and the file names are listed in a column in the CSV.

Create a Collection in Islandora 2.0

Islandora Workbench works best when ingesting one collection at a time. To begin, login to Islandora 2.0 in your web browser and create a new collection.

Get the CSV File

Islandora Workbench requires a csv in either Google Sheets or on your local disk. The AG_Photos spreadsheet is provided as a sample input_csv and can be upload to your Google Drive

[AG_Photos.xlsx](#)

Prepare Config File

Islandora Workbench uses YAML files to configure its operations. These files are [documented in detail](#). Here is an example config file, including a link to a sample CSV. You must download the CSV and open in Google Sheets to be able to correctly run the example.

```
task: create
host: "https://islandora.traefik.me/"
username: xxxx
password: xxxx
media_type: file
input_csv: 'xxx'
id_field: PID
csv_field_templates:

  - field_rights: "http://rightsstatements.org/vocab/CNE/1.0/"
  - field_member_of: xxxx
  - field_model: xxxx
  - field_resource_type: xxxx
  - field_display_hints: xxxx

default_file_mimetype: 'image/tiff'
default_file_extension: ".tif"
use_node_title_for_media: 1
allow_adding_terms: true
```

NOTE: This CSV associated with this config file uses file URLs. To use a file directory, the input_dir configuration option may be used. More information is available in the [Workbench documentation](#).

The user credentials you include must be for a user who has permission to create objects and taxonomy terms.

The **csv_field_templates** are fields that will apply to every resource in the collection. The numbers referenced in these fields are Drupal Node IDs; you will need to update these numbers in your config file based on the Node IDs in your Drupal instance.

input_csv

The public link to your spreadsheet in Google Sheets

Note: If the gid of your spreadsheet does not automatically set to 0, you may need to set **google_sheets_gid** with the value from your spreadsheet. More information is available in the relevant [workbench documentation](#).

field_member_of

This is the Node ID of the collection you created in step 2. You can find the ID by hovering over any of the tabs when you view the collection - it will be in the URL as "/node/id".

field_model

The ID of the Islandora Model used by items in this collection. You can find a list of models and associated Node IDs by going to https://your.site/admin/structure/taxonomy/manage/islandora_models/overview*. In this case, this is a collection of images, so we will go with the Image model.

**Note: This link is to indicate the path structure for your own specific site. You should replace "your.site" in the above listed URL with your actual Islandora site URL.*

field_resource_type

The ID of the resource type used by items in this collection. This is likely to be similar to the Islandora Model used above. You can find a list of resource types and associated IDs by going to https://your.site/admin/structure/taxonomy/manage/resource_types/overview. This collection uses the Image resource type.

field_display_hints

Display hints are used to indicate where a viewer should be used. You can find the list of display hints and associated IDs at https://your.site/admin/structure/taxonomy/manage/islandora_display/overview. These are large images so we'll want to use the Open Seadragon viewer.

Prepare CSV File

CSV Required Fields

The CSV must include the following required columns:

- Title
- ID - this is only used by Workbench and is not migrated into the Repository Item node.
- Resource Type
- System Model
- File - a path to the media file (if applicable)

Title

This will be the header on the object page, and will display as the object title on collection and search results pages.

Islandora Workbench supports non-Latin characters in CSV, provided the CSV file is encoded as ASCII or UTF-8.

Drupal's maximum allowed length is 255 characters. If some of your object titles are longer than that, we may want to install a Drupal module that allows us to exceed that limit (e.g., [Node Title Length](#) or [Entity Title Length](#)).

Resource Type

Default fields are:

Term Name	External URI
Collection	http://purl.org/dc/dcmitype/Collection
Dataset	http://purl.org/dc/dcmitype/Dataset
Image	http://purl.org/dc/dcmitype/Image
Interactive Resource	http://purl.org/dc/dcmitype/InteractiveResource
Moving Image	http://purl.org/dc/dcmitype/MovingImage
Physical Object	http://purl.org/dc/dcmitype/PhysicalObject
Service	http://purl.org/dc/dcmitype/Service
Software	http://purl.org/dc/dcmitype/Software
Sound	http://purl.org/dc/dcmitype/Sound
Still Image	http://purl.org/dc/dcmitype/StillImage
Text	http://purl.org/dc/dcmitype/Text

New terms can be created on the fly (during ingest). If your metadata uses terms that already exist by default, you can reference the term name, its ID, or its URI (if it has one) in the CSV.

NOTE: The Born-Digital i8 theme requires specific combinations of Resource Type and Model terms in order for compound objects, collections, and paged objects to display correctly. Please refer to [Appendix B: Born-Digital i8 Theme Object View Configurations](#).

System Model

Available terms are:

Term Name	External URI
Audio	http://purl.org/coar/resource_type/c_18cc
Binary	http://purl.org/coar/resource_type/c_1843
Collection	http://purl.org/dc/dcmitype/Collection#Model
Compound Object	http://vocab.getty.edu/aat/300242735
Digital Document	https://schema.org/DigitalDocument
Image	http://purl.org/coar/resource_type/c_c513
Newspaper	https://schema.org/Newspaper
Page	http://id.loc.gov/ontologies/bibframe/part

Paged Content	https://schema.org/Book
Publication Issue	https://schema.org/PublicationIssue
Video	http://purl.org/coar/resource_type/c_12ce

NOTE: The Born-Digital i8 theme requires specific combinations of Resource Type and Model terms in order for compound objects, collections, and paged objects to display correctly. Please refer to [Appendix B: Born-Digital i8 Theme Object View Configurations](#).

File

The following comes from the [Workbench Documentation](#):

Values in the file field contain the location of files that are used to create Drupal Media. Workbench can create only one media per CSV record. ... File locations can be relative to the directory named in input_dir [in the .yaml file], absolute paths, or URLs. Examples of each:

- **relative to directory named in the input_dir configuration setting:** myfile.png
- **absolute:** /tmp/data/myfile.png
- **URL:** <http://example.com/files/myfile.png>

Things to note about file values in general:

- Relative, absolute, and URL file locations can exist within the same CSV file.
- By default, if the file value for a row is empty, Workbench's --check option will show an error. But, in some cases you may want to create nodes but not add any media. If you add allow_missing_files: true to your config file for "create" tasks, you can leave the file column in your CSV empty.
- If you do not want to create media for any of the rows in your CSV file, you can include nodes_only: true in your configuration file.
- Currently, file values can only contain characters in the ASCII or Latin-1 character sets. The following characters with diacritics should be safe in filenames: Å Ä Å Ä Å Æ Ç È É Ê Ë Ì Í Î Ï Ð Ñ Ò Ó Ô Õ Ö × Ù Ú Û Ü Ý Þ ß à á â ã ä å æ ç è é ê ë ì í î ï ð ñ ò ó ô õ ö ÷ ø ù ú û ü ý þ ÿ.

Things to note about URLs as file values:

- Workbench downloads files identified by URLs and saves them in the directory named in input_dir [in the .yaml file] before processing them further; within this directory, each file is saved in a subdirectory named after the value in the row's id_field field. It does not delete the files from these locations after they have been ingested into Islandora unless the delete_tmp_upload configuration option [in the .yaml file] is set to true.
- Files identified by URLs must be accessible to the Workbench script, which means they must not require a username/password; however, they can be protected by a firewall, etc. as long as the computer running Workbench is allowed to retrieve the files without authenticating.
- Currently Workbench requires that the URLs point directly to a file or a service that generates a file, and not a wrapper page or other indirect route to the file.

Other General CSV Notes

Delimiter

The default delimiter is , [comma] but this can be configured in the Workbench .yaml file.

Subdelimiter

The default subdelimiter, to indicate separation between multiple values in one cell, is | [pipe] but this can be configured in the Workbench .yaml file.

Term Creation

Workbench will create new vocabulary terms on the fly (they do not need to already be in Drupal), as long as this requirement is specified in the .yaml file. But note the following:

- If a term name is longer than 255 characters, Workbench will truncate it at that length, log that it has done so, and create the term.
- Creating taxonomy terms by including them in your CSV file adds new terms to the root of the applicable vocabulary. Workbench cannot create a new term that has another term as its parent (i.e., terms below the top level of a hierarchical taxonomy). However, for existing terms, Workbench doesn't care where they are in a taxonomy's hierarchy.
- Taxonomy terms created with new nodes **are not removed when you delete the nodes**.

Fields Created by Default

The following fields are available to be used as columns in your CSV by default. If additional fields are needed, these will need to be added manually (see Adding New Fields below). Not all fields must be used; only fields with data in them will be displayed on the object page.

The **machine names** for each field are what need to be used as the column headers in the CSV. To find the machine name for a field, go to Structure > Content Type > Repository Item > Manage Fields.

Label (displayed on the front-end)	Machine Name or Workbench-required name (use for column header in CSV)	Field Type and Notes
Title	title	text field (see more information above, under CSV Required Fields)
Alternative Title	field_alternative_title	text field
Identifier	field_identifier	text field; multi-value
Resource Type	field_resource_type	taxonomy reference (see more information above, under CSV Required Fields)
Genre	field_genre	taxonomy reference; multi-value
Linked Agent	field_linked_agent	typed relation field; multi-value - see more information about how to set up this field below, under Linked Agent.
Date Created	field_edtf_date_created	EDTF field; multi-value; must be in EDTF format - see more information about how to set up this field below, under EDTF Formats.
Date Issued	field_edtf_date_issued	EDTF field; multi-value; must be in EDTF format - see more information about how to set up this field below, under EDTF Formats.
Date	field_edtf_date	EDTF field; multi-value; must be in EDTF format - see more information about how to set up this field below, under EDTF Formats.
Edition	field_edition	text field; multi-value
Place Published	field_place_published	text field; multi-value
Language	field_language	taxonomy reference field; multi-value
Description	field_description_long	text-formatted-long, can support line breaks If your metadata has line breaks , as long as they are included in the cell in the spreadsheet, saving as a CSV should wrap the content of the cell in quotes and the line break will be preserved.
Table of Contents	field_table_of_contents	text-formatted-long, can support line breaks
Physical Form	field_physical_form	taxonomy reference; multi-value
Extent	field_extent	text field; multi-value
Rights	field_rights	text field; multi-value; if you want this to be an external link, field needs to be changed to a Link type field.
Subject	field_subject	taxonomy reference (from corporate body, family, geographic location, person, or subject vocabularies); multi-value; data in CSV must include a namespace before the term (i.e., person: , corporate_body: , geo_location: , family: , or subject:).
Geographic Subject	field_geographic_subject	taxonomy reference (from Geographic Location vocabulary); multi-value
Coordinates	field_coordinates	geolocation fields (latitude and longitude); multi-value - see more information about how to set up this field below, under Coordinates
Coordinates (Text)	field_coordinates_text	text field; multi-value
Temporal Subject	field_temporal_subject	taxonomy reference (from Temporal vocabulary); multi-value
Subjects (name)	field_subjects_name	taxonomy reference (from person, family, or corporate body); multi-value; data in CSV must include a namespace before the term (i.e., person: , family: , or corporate_body)
Dewey Classification	field_dewey_classification	text field; multi-value
Library of Congress Classification	field_lcc_classification	text field; multi-value
Classification (Text)	field_classification	text field; multi-value

Local Identifier	field_local_identifier	text field; multi-value
ISBN	field_isbn	text field; multi-value
OCLC Number	field_oclc_number	text field; multi-value
Note	field_note	text-formatted-long, can support line breaks; multi-value
System Model	field_model	taxonomy reference (from Islandora Models vocabulary); (see more information above, under CSV Required Fields)
Member of	field_member_of	node reference; multi-value; points to the containing collection or parent node - see more information below under Member Of
Language	n/a	(dropdown; defaults to English - ignore this for the purposes of the CSV)
Access Control	field_access_terms	taxonomy reference (from Islandora Access vocabulary)
Display hints	field_display_hints	taxonomy reference (from Islandora Display vocabulary) - see more information below, under Display Hints
PID	field_pid	text field
Weight	field_weight	number (integer); indicates the order of a resource in a collection of resources (used for compound objects and paged content)

Adding New Fields

Follow these steps to add new fields as required:

1. Navigate to Structure > Content types > Repository item > Manage fields
2. Click Add Field
3. Select the Field Type based on your requirements
 - a. Note: For Entity Reference field types you will need to select Taxonomy Term under Typed relation when setting the field type.
4. Add a Label based on the table
5. Save the Field settings
6. For Entity Reference and Typed Relation fields:
 - a. Check the "Create referenced entities if they don't already exist" box
 - b. Select the appropriate vocabulary (or vocabularies) based on the table

NOTE: If additional taxonomies are required for any of the new fields, these should be created prior to creating the new fields. To create a new taxonomy, follow these steps:

1. Go to Structure > Taxonomy to view existing vocabularies.
2. Click Add Vocabulary to create the desired vocabulary
3. Give it a label and click Save
4. Click Add Term to populate the list, or leave it blank to be filled automatically during ingest
5. Follow the same steps to create any remaining vocabularies.

Special Fields and Field Types

Linked Agent

A Relationship Type and either family, person, or corporate_body must be specified. The list of Relationship Types available by default is shown in [Appendix A: Default Relationship Types](#). Others can be added, but it requires customization.

Linked Agents can be Person, Family, or Corporate Body, by default. They must be one of these (or if they must reference another vocabulary, let Born-Digital know).

The data in the cell must include the word "relators" as the namespace, followed by the abbreviation for the applicable Relationship Type (as shown in Appendix A), followed by either person, family, or corporate_body, followed by the name.

Multiple values (of different Relationship Types and person/corporate body/family) can be in one cell, separated by a | [pipe].

Example 1:

relators:cre:person:Poole, A. F.|relators:cre:corporate_body:Beck & Pauli

This shows a person whose relationship type (or role) is "Creator" and a corporate body whose relationship type (or role) is "Creator."

Example 2:

relators:cre:person:Peter Boesman|relators:pbl:corporate_body:xeno-canto

This shows a person whose relationship type (or role) is “Creator” and a corporate body whose relationship type (or role) is “Publisher.”

Adding a Custom Relation

Relationships that are not available by default, per Appendix A, can be added:

1. Go to Structure > Content Types. Scroll to “Repository Item” and click “Manage Fields.”
2. Find the Linked Agent field and click “Edit.”
3. Scroll down to “Available Relations” field and add the new relation. For example: local:dpt|Department (dpt)
4. Click “Save Settings”.
5. In your CSV, you will reference the new relation just like the others - for example: local:dpt:corporate_body:Test Department.

NOTE: This will NOT enable you to add the new relationship as a facet. That requires custom development work (the relationship needs to be added to the module that the theme uses to provide facets).

EDTF Formats

All dates must follow [EDTF formatting rules](#). Here are some examples:

EDTF Input	Front-End Output
1933?	1933 (year uncertain)
1945~	1945 (year approximate)
2016-04-12	2016-04-12
1860/1880?	1860 to 1880 (year uncertain)
1870/1880	1870 to 1880

Links

A Link field type stores URLs and link text in separate data elements.

The following comes from the [Workbench Documentation](#):

To add or update fields of this type, Workbench needs to provide the URL and link text in the structure Drupal expects. To accomplish this within a single CSV field, we separate the URL and link text pairs in CSV values with double percent signs (%%), like this:

```
field_related_websites
http://acme.com%%Acme Products Inc.
```

You can include multiple pairs of URL/link text pairs in one CSV field if you separate them with the subdelimiter character:

```
field_related_websites
http://acme.com%%Acme Products Inc.|http://diy-first-aid.net%%DIY First Aid
```

The URL is required, but the link text is not. If you don't have or want any link text, omit it and the double percent signs:

```
field_related_websites
http://acme.com

field_related_websites
http://acme.com|http://diy-first-aid.net%%DIY First Aid
```

Coordinates

The Coordinates field uses the Geolocation field type.

The following comes from the [Workbench Documentation](#):

The Geolocation field type, managed by the Geolocation Field contrib module, stores latitude and longitude coordinates in separate data elements. To add or update fields of this type, Workbench needs to provide the latitude and longitude data in these separate elements.

To simplify entering geocoordinates in the CSV file, Workbench allows geocoordinates to be in lat,long format, i.e., the latitude coordinate followed by a comma followed by the longitude coordinate. When Workbench reads your CSV file, it will split data on the comma into the required lat and long parts. An example of a single geocoordinate in a field would be:

```
field_coordinates
"49.16667,-123.93333"
```

You can include multiple pairs of geocoordinates in one CSV field if you separate them with the subdelimiter character:

```
field_coordinates
"49.16667,-123.93333|49.25,-124.8"
```

Note that:

- Geocoordinate values in your CSV need to be wrapped in double quotation marks, unless the delimiter key in your configuration file is set to something other than a comma.
- If you are entering geocoordinates into a spreadsheet, a leading + will make the spreadsheet application think you are entering a formula. You can work around this by escaping the + with a backslash (\), e.g., 49.16667,-123.93333 should be \+49.16667,-123.93333, and 49.16667,-123.93333|49.25,-124.8 should be \+49.16667,-123.93333|\+49.25,-124.8. Workbench will strip the leading \ before it populates the Drupal fields.

Member Of

The “Member of” field determines the parent of the object. It is used to identify the collection to which an object belongs, or the parent/container object if the object is a page or compound object.

For the purposes of Workbench, the Member Of column is used **if you have a pre-existing collection in Drupal** into which you want to ingest an object. In this case, you will enter the collection’s node ID in the Member Of column in the object row that belongs to the collection.

Page objects will always be “Member of” > the newspaper issue (System Model=”Publication Issue”) to which they belong, **or** the book (System Model=”Paged Content”) to which they belong.

Newspaper issues will always be “Member of” > the newspaper parent (System Model=”Newspaper”) to which they belong.

Compound child objects will always be “Member of” > the compound object parent (System Model=”Compound Object”) to which they belong.

Display Hints

These terms, from the Islandora Display vocabulary, will define which viewer is used on an object page.

Term Name	Used For
Open Seadragon	Large Image, Page
PDFjs	PDF

Configure the CSV

Your CSV will include only columns that are a) required and b) have data in them. You should not include columns in your CSV that do not have data. This will cause an error during Workbench’s configuration check process.

If the collection(s) that will contain the objects have NOT yet been created in Drupal, you can include rows for the collection(s) in your CSV. Each object will also have a row, and will reference the id of the collection it belongs to. For example:

title	id	parent_id	field_member_of
Test Collection	55		
Easthampton Town Hall	1	55	
Nehemiah Strong House	2	55	
Amherst College, Lawrence Observatory	3	55	

If the collection(s) that will contain the objects already exist in Drupal, you can use the **field_member_of** column to reference the node ID of the collection (s). (In this case, you will not use the parent_id column except for compound objects and paged content; see [Configuring Complex Objects in the CSV](#) for more information.)

To find the collection’s node ID:

1. Click on “Content” in the admin menu, to go to ../admin/content.
2. Find the collection node in the table of nodes.
3. In the far right column of the table, hover on the “edit” link.
4. Look at the bottom of your screen and you will see a URL that includes ../node/XX - e.g., ../node/103. The number following /node/ is the node ID. This is the number you will reference in the field_member_of column of your CSV.

Assuming, for example, your top-level collection had a node ID of 100, your CSV would look like this (this is only showing a portion of the columns):

title	id	parent_id	field_member_of
Easthampton Town Hall	1		100
Nehemiah Strong House	2		100

Amherst College, Lawrence Observatory	3	100
---------------------------------------	---	-----

Configuring Complex Objects in the CSV

Newspapers

IMPORTANT: Newspaper Issues need a Date Issued (and this field needs to be set to be visible to anonymous users) in order to be displayed correctly in the Newspaper Parent view.

A Newspaper will generally consist of three parts:

1. **Newspaper Parent** - the node that will be the parent of all associated Newspaper Issues. This node will not have any media associated with it.
2. **Newspaper Issue** - the node that will be a child of the Newspaper Parent and that will be the parent of all associated Newspaper Pages. The only media file that might be associated with this node would be a PDF compilation of its pages.
3. **Newspaper Page(s)** - the node(s) that will be the child(ren) of the Newspaper Issue and that will reference the media files that comprise the Newspaper Issue. This includes images as well as extracted text file derivatives.

Assuming your Newspaper Parent node has not yet been created, and you'd like it to be contained within another top-level collection whose node has also not yet been created, the table below shows how the three newspaper parts (and the top-level collection) would be laid out in your CSV:

title	id	parent_id	field_weight
Sample Collection	1		
Connecticut Western News (Newspaper)	2	1	
Connecticut Western News Vol. 1 No. 7	3	2	
Connecticut Western News Vol. 1 No. 7 - page 1	4	3	1
Connecticut Western News Vol. 1 No. 7 - page 2	5	3	2
Connecticut Western News Vol. 1 No. 7 - page 3	6	3	3
Connecticut Western News Vol. 1 No. 7 - page 4	7	3	4

The Newspaper Parent is a child of the containing collection (or it may have no parent). The Newspaper Issue is a child of the Newspaper Parent, and each Newspaper Page is a child of the Newspaper Issue.

The Newspaper Pages must each have a **field_weight** assigned.

In this case, there would be no data in the **field_member_of** column of your CSV. All parent/child relationships are defined using the **parent_id** column.

If, however, your Newspaper Parent is already present on your site, you can use **field_member_of** to identify the parent using its node ID. In that case, the Newspaper Issue row would have nothing in the **parent_id** column, but would include the Newspaper Parent's node ID in the **field_member_of** column. The Newspaper Page rows would have the Newspaper Issue's id in the **parent_id** column. These rows of your CSV would look like this (assuming the Newspaper Parent's node ID were "100"):

title	id	parent_id	field_weight	field_member_of
Connecticut Western News Vol. 1 No. 7	1			100
Connecticut Western News Vol. 1 No. 7 - page 1	2	1	1	
Connecticut Western News Vol. 1 No. 7 - page 2	3	1	2	
Connecticut Western News Vol. 1 No. 7 - page 3	4	1	3	
Connecticut Western News Vol. 1 No. 7 - page 4	5	1	4	

Books

Configuring Books in your CSV is very similar to configuring Newspapers. The parent Book object will have child Page objects.

A Book will generally consist of two parts:

1. **Book parent** - the node that will be the parent of all associated book pages. This node will not have any media associated with it.
2. **Book Page(s)** - the node(s) that will be the child(ren) of the Book and that will reference the media files that comprise the Book. This includes images as well as extracted text file derivatives.

Assuming your Book Parent node has not yet been created, and you'd like it to be contained within a collection whose node has also not yet been created, the table below shows how the two Book parts (and the collection) would be laid out in your CSV:

title	id	parent_id	field_weight
Sample Collection	1		
On the Tides at Malta (Book)	2	1	
On the Tides at Malta - page 1	3	2	1
On the Tides at Malta - page 2	4	2	2
On the Tides at Malta - page 3	5	2	3
On the Tides at Malta - page 4	6	2	4
On the Tides at Malta - page 5	7	2	5

The Book Pages must each have a **field_weight** assigned.

In this case, there would be no data in the **field_member_of** column of your CSV. All parent/child relationships are defined using the **parent_id** column.

If, however, the collection into which you are ingesting the Book is already present on your site, you can use **field_member_of** to identify the parent using its node ID. In that case, the Book Parent would not have anything in the **parent_id** column, but would include the collection node ID in the **field_member_of** column. The Book Page rows would have the Book Parent's id in the **parent_id** column. These rows of your CSV would look like this (assuming the collection node ID were "100"):

title	id	parent_id	field_weight	field_member_of
On the Tides at Malta (Book)	1			100
On the Tides at Malta - page 1	2	1	1	
On the Tides at Malta - page 2	3	1	2	
On the Tides at Malta - page 3	4	1	3	
On the Tides at Malta - page 4	5	1	4	
On the Tides at Malta - page 5	6	1	5	

Compound Objects

Configuring Compound Objects in your CSV is very similar to Newspapers and Books. The Compound Object will have a containing "parent object" as well as one or more children. The containing parent object will not be visible on the front-end, aside from its metadata (it cannot have any media of its own; that would never be seen).

When viewing Compound Objects on the site, you will view one child object at a time, with a viewer containing the media of the child object and a gallery of thumbnails for all child objects below. You will see two tabs, one showing the metadata associated with the child object you're viewing, and one showing the metadata associated with the containing parent object.

A Compound Object will consist of two parts:

1. **Parent** - the node that will be the parent of all associated child objects. This node will not have any media associated with it, but can contain metadata.
2. **Children** - the node(s) that will be the child(ren) of the parent and that contain media files and metadata.

Assuming you'd like the Compound Object to be contained within a collection whose node has also not yet been created, the table below shows how the two Compound Object parts (and the collection) would be laid out in your CSV:

title	id	parent_id	field_weight
Sample Collection	1		
Historic Western Mass (Compound Object)	2	1	
Amherst 1886 (Child Object 1)	3	2	1
Adams 1882 (Child Object 2)	4	2	2

The child objects must each have a **field_weight** assigned.

In this case, there would be no data in the **field_member_of** column of your CSV. All parent/child relationships are defined using the **parent_id** column.

If, however, the collection into which you are ingesting the Compound Object is already present on your site, you can use **field_member_of** to identify the parent using its node ID. In that case, the Compound Object parent would not have anything in the **parent_id** column, but would include the collection node ID in the **field_member_of** column. The child object rows would have the Compound Object parent's id in the **parent_id** column. These rows of your CSV would look like this (assuming the collection node ID were "100"):

title	id	parent_id	field_weight	field_member_of
Historic Western Mass (Compound Object)	1			100
Amherst 1886 (Child Object 1)	2	1	1	
Adams 1882 (Child Object 2)	3	1	2	

Check, Then Run

You should always [check your configuration and spreadsheet are valid](#) before running the ingest. Fortunately, Islandora Workbench makes this easy with the `--check` command:

```
./workbench --config config.yml --check
```

The check command will report out any errors so you can fix them before running the ingest.

Once no more errors are present, simply run the same command without `--check`:

```
./workbench --config config.yml
```

Quality Assurance/Quality Control post-ingest

After a collection has been ingested, it is important to check that all objects appear and function as expected. Careful quality control *before ingest* of the spreadsheet data and column headings, together with the Workbench validity checks, can help to prevent many common errors.

Common errors included:

- Errors in taxonomy terms or other metadata due to mistakes in spreadsheet formatting or incorrect (not updated) Workbench settings
- Objects not appearing in the correct collection, presumably due to an error in the `member_of` field on ingest
- Metadata appearing in the incorrect field due to inconsistencies between column headers and column contents
- Non-generation of thumbnails, resulting in child objects not being visible from the parent object page. Thumbnails are supposed to be generated on ingest of an original file; in the case of some large PDFs, that generation process seems to have stalled.

Mitigation strategies might include programmatic fixes to taxonomy terms; searching for missing objects by name or associated metadata term and reassigning the `member_of` field by hand (such a search should always be done before reingesting objects); remediating metadata by hand or, for entire collections, via Workbench; generating thumbnails using Drupal actions, or uploading thumbnails by hand (if thumbnails are available from an Islandora 7 site, they may be downloaded from the object TN datastream and uploaded to Islandora 2.0 object media).

It may be possible to semi-automate some quality checks by, for example, configuring a view that can show media that do not have an associated thumbnail.

Appendix A: Default Relationship Types

relators:abr|Abridger (abr)

relators:act|Actor (act)

relators:adp|Adapter (adp)

relators:rcp|Addressee (rcp)

relators:anl|Analyst (anl)

relators:anm|Animator (anm)

relators:ann|Annotator (ann)

relators:apl|Appellant (apl)

relators:ape|Appellee (ape)
relators:app|Applicant (app)
relators:arc|Architect (arc)
relators:arr|Arranger (arr)
relators:acp|Art copyist (acp)
relators:adi|Art director (adi)
relators:art|Artist (art)
relators:ard|Artistic director (ard)
relators:asg|Assignee (asg)
relators:asn|Associated name (asn)
relators:att|Attributed name (att)
relators:auc|Auctioneer (auc)
relators:aut|Author (aut)
relators:aqt|Author in quotations or text abstracts (aqt)
relators:aft|Author of afterword, colophon, etc. (aft)
relators:aud|Author of dialog (aud)
relators:auil|Author of introduction, etc. (auil)
relators:ato|Autographer (ato)
relators:ant|Bibliographic antecedent (ant)
relators:bnd|Binder (bnd)
relators:bdd|Binding designer (bdd)
relators:blw|Blurb writer (blw)
relators:bkd|Book designer (bkd)
relators:bkp|Book producer (bkp)
relators:bjd|Bookjacket designer (bjd)
relators:bpd|Bookplate designer (bpd)
relators:bsl|Bookseller (bsl)
relators:brl|Braille embosser (brl)
relators:brd|Broadcaster (brd)
relators:cil|Calligrapher (cil)
relators:ctg|Cartographer (ctg)
relators:cas|Caster (cas)
relators:cns|Censor (cns)
relators:chr|Choreographer (chr)
relators:clb|Collaborator (clb; deprecated, use Contributor)
relators:cng|Cinematographer (cng)
relators:cli|Client (cli)
relators:cor|Collection registrar (cor)
relators:col|Collector (col)
relators:clt|Collotyper (clt)

relators:clr|Colorist (clr)
relators:cmm|Commentator (cmm)
relators:cwt|Commentator for written text (cwt)
relators:com|Compiler (com)
relators:cpl|Complainant (cpl)
relators:cpt|Complainant-appellant (cpt)
relators:cpe|Complainant-appellee (cpe)
relators:cmp|Composer (cmp)
relators:cmt|Compositor (cmt)
relators:ccp|Conceptor (ccp)
relators:cnd|Conductor (cnd)
relators:con|Conservator (con)
relators:csl|Consultant (csl)
relators:csp|Consultant to a project (csp)
relators:cos|Contestant (cos)
relators:cot|Contestant-appellant (cot)
relators:coe|Contestant-appellee (coe)
relators:cts|Contestee (cts)
relators:ctt|Contestee-appellant (ctt)
relators:cte|Contestee-appellee (cte)
relators:ctr|Contractor (ctr)
relators:ctb|Contributor (ctb)
relators:cpc|Copyright claimant (cpc)
relators:cph|Copyright holder (cph)
relators:crr|Corrector (crr)
relators:crp|Correspondent (crp)
relators:cst|Costume designer (cst)
relators:cou|Court governed (cou)
relators:crt|Court reporter (crt)
relators:cov|Cover designer (cov)
relators:cre|Creator (cre)
relators:cur|Curator (cur)
relators:dnc|Dancer (dnc)
relators:dte|Data contributor (dte)
relators:dtm|Data manager (dtm)
relators:dte|Dedicatee (dte)
relators:dto|Dedicator (dto)
relators:dfd|Defendant (dfd)
relators:dft|Defendant-appellant (dft)
relators:dfe|Defendant-appellee (dfe)

relators:dgg|Degree granting institution (dgg)
relators:dgs|Degree supervisor (dgs)
relators:dln|Delineator (dln)
relators:dpc|Depicted (dpc)
relators:dpt|Depositor (dpt)
relators:dsr|Designer (dsr)
relators:drt|Director (drt)
relators:dis|Dissertant (dis)
relators:dbp|Distribution place (dbp)
relators:dst|Distributor (dst)
relators:dnr|Donor (dnr)
relators:drm|Draftsman (drm)
relators:dub|Dubious author (dub)
relators:edt|Editor (edt)
relators:edc|Editor of compilation (edc)
relators:edm|Editor of moving image work (edm)
relators:elg|Electrician (elg)
relators:elt|Electrotypist (elt)
relators:enj|Enacting jurisdiction (enj)
relators:eng|Engineer (eng)
relators:egr|Engraver (egr)
relators:etr|Etcher (etr)
relators:evp|Event place (evp)
relators:exp|Expert (exp)
relators:fac|Facsimilist (fac)
relators:fld|Field director (fld)
relators:fmd|Film director (fmd)
relators:fds|Film distributor (fds)
relators:flm|Film editor (flm)
relators:fmp|Film producer (fmp)
relators:fmk|Filmmaker (fmk)
relators:fpv|First party (fpv)
relators:frg|Forger (frg)
relators:fmo|Former owner (fmo)
relators:fnd|Funder (fnd)
relators:gis|Geographic information specialist (gis)
relators:grt|Graphic technician (grt; deprecated, use Artist)
relators:hnr|Honoree (hnr)
relators:hst|Host (hst)
relators:his|Host institution (his)

relators:ilu|Illuminator (ilu)
relators:ill|Illustrator (ill)
relators:ins|Inscriber (ins)
relators:itr|Instrumentalist (itr)
relators:ive|Interviewee (ive)
relators:ivr|Interviewer (ivr)
relators:inv|Inventor (inv)
relators:isb|Issuing body (isb)
relators:jud|Judge (jud)
relators:jug|Jurisdiction governed (jug)
relators:lbr|Laboratory (lbr)
relators:ldr|Laboratory director (ldr)
relators:lsa|Landscape architect (lsa)
relators:led|Lead (led)
relators:len|Lender (len)
relators:lil|Libelant (lil)
relators:lit|Libelant-appellant (lit)
relators:lie|Libelant-appellee (lie)
relators:lel|Libelee (lel)
relators:let|Libelee-appellant (let)
relators:lee|Libelee-appellee (lee)
relators:lbt|Librettist (lbt)
relators:lse|Licensee (lse)
relators:iso|Licensor (iso)
relators:lgd|Lighting designer (lgd)
relators:ltg|Lithographer (ltg)
relators:lyr|Lyricist (lyr)
relators:mfp|Manufacture place (mfp)
relators:mfr|Manufacturer (mfr)
relators:mrb|Marbler (mrb)
relators:mrk|Markup editor (mrk)
relators:med|Medium (med)
relators:mdc|Metadata contact (mdc)
relators:mte|Metal-engraver (mte)
relators:mtk|Minute taker (mtk)
relators:mod|Moderator (mod)
relators:mon|Monitor (mon)
relators:mcp|Music copyist (mcp)
relators:msd|Musical director (msd)
relators:mus|Musician (mus)

relators:nrt|Narrator (nrt)
relators:osp|Onscreen presenter (osp)
relators:opn|Opponent (opn)
relators:orm|Organizer (orm)
relators:org|Originator (org)
relators:oth|Other (oth)
relators:own|Owner (own)
relators:pan|Panelist (pan)
relators:ppm|Papermaker (ppm)
relators:pta|Patent applicant (pta)
relators:pth|Patent holder (pth)
relators:pat|Patron (pat)
relators:prf|Performer (prf)
relators:pma|Permitting agency (pma)
relators:pht|Photographer (pht)
relators:ptf|Plaintiff (ptf)
relators:ptt|Plaintiff-appellant (ptt)
relators:pte|Plaintiff-appellee (pte)
relators:plt|Platemaker (plt)
relators:pra|Praeses (pra)
relators:pre|Presenter (pre)
relators:prt|Printer (prt)
relators:pop|Printer of plates (pop)
relators:prm|Printmaker (prm)
relators:prc|Process contact (prc)
relators:pro|Producer (pro)
relators:prn|Production company (prn)
relators:prs|Production designer (prs)
relators:pmn|Production manager (pmn)
relators:prd|Production personnel (prd)
relators:prp|Production place (prp)
relators:prg|Programmer (prg)
relators:pdr|Project director (pdr)
relators:pfr|Proofreader (pfr)
relators:prv|Provider (prv)
relators:pup|Publication place (pup)
relators:pbl|Publisher (pbl)
relators:pbd|Publishing director (pbd)
relators:ppt|Puppeteer (ppt)
relators:rdd|Radio director (rdd)

relators:rpc|Radio producer (rpc)
relators:rce|Recording engineer (rce)
relators:rcd|Recordist (rcd)
relators:red|Redaktor (red)
relators:ren|Renderer (ren)
relators:rpt|Reporter (rpt)
relators:rps|Repository (rps)
relators:rth|Research team head (rth)
relators:rtm|Research team member (rtm)
relators:res|Researcher (res)
relators:rsp|Respondent (rsp)
relators:rst|Respondent-appellant (rst)
relators:rse|Respondent-appellee (rse)
relators:rpv|Responsible party (rpv)
relators:rsg|Restager (rsg)
relators:rsr|Restorationist (rsr)
relators:rev|Reviewer (rev)
relators:rbr|Rubricator (rbr)
relators:sce|Scenarist (sce)
relators:sad|Scientific advisor (sad)
relators:aus|Screenwriter (aus)
relators:scr|Scribe (scr)
relators:scl|Sculptor (scl)
relators:spy|Second party (spy)
relators:sec|Secretary (sec)
relators:sll|Seller (sll)
relators:std|Set designer (std)
relators:stg|Setting (stg)
relators:sgn|Signer (sgn)
relators:sng|Singer (sng)
relators:sds|Sound designer (sds)
relators:spk|Speaker (spk)
relators:spn|Sponsor (spn)
relators:sgd|Stage director (sgd)
relators:stm|Stage manager (stm)
relators:stn|Standards body (stn)
relators:str|Stereotyper (str)
relators:stl|Storyteller (stl)
relators:shh|Supporting host (shh)
relators:svr|Surveyor (svr)

relators:tch|Teacher (tch)
 relators:tcd|Technical director (tcd)
 relators:tld|Television director (tld)
 relators:tlp|Television producer (tlp)
 relators:ths|Thesis advisor (ths)
 relators:trc|Transcriber (trc)
 relators:trl|Translator (trl)
 relators:tyd|Type designer (tyd)
 relators:tyg|Typographer (tyg)
 relators:uvp|University place (uvp)
 relators:vdg|Videographer (vdg)
 relators:voc|Vocalist (voc; deprecated, use Singer)
 relators:vac|Voice actor (vac)
 relators:wit|Witness (wit)
 relators:wde|Wood engraver (wde)
 relators:wdc|Woodcutter (wdc)
 relators:wam|Writer of accompanying material (wam)
 relators:wac|Writer of added commentary (wac)
 relators:wal|Writer of added lyrics (wal)
 relators:wat|Writer of added text (wat)
 relators:win|Writer of introduction (win)
 relators:wpr|Writer of preface (wpr)
 relators:wst|Writer of supplementary textual content (wst)

Appendix B: Born-Digital i8 Theme Object View Configurations

Object Type	Resource Type term	System Model term
Collection	Collection	Collection
Audio	Sound	Audio
Basic Image (jpg)	Still Image	Image
Binary	[any]	Binary
Book (parent of pages)	Collection	Paged Content
Compound Object	Collection	Compound Object
Large Image (tiff)	Still Image	Image
Newspaper (parent of issues)	Collection	Newspaper
Newspaper Issue (parent of pages)	Collection	Publication Issue
Newspaper Page	Text	Page
Page	Text	Page
PDF	[any]	Digital Document

Video	Moving Image	Video
-------	--------------	-------